

CHUHAO (JACKY) ZENG

SUMMARY OF QUALIFICATIONS:

- Strong academic background in computer science
- Solid mathematical abilities (Calculus, Linear Algebra, Mathematics for Computer Science, Abstract Mathematics, Number Theory, Combinatorics, algorithms)
- Extensive training in data analysis (Statistics, Statistics for Computer Science, Design of Scientific Studies, Surveys Sampling and Observational Data, Methods of Data Analysis)
- Technical skills: C, Java, Python, HTML, Haskell, MATLAB, HTML, JavaScript
-

EDUCATION:

B.Sc. in Computer Science (Specialist Degree)

University of Toronto Toronto, Canada

Sept. 2016 – May 2019

- CGPA 3.67/4.00 (High Distinction)
- Awards: Dean's List (2018 Winter; 2018 Summer)
- Relevant coursework: Computer Programming, Software design, Software Tools and Systems Programming, Theory of Computation, Data Structures and Analysis, Algorithm Design and Analysis, Numerical Methods

IN COURSE PROJECTS

Crossroad Simulation System Using Locks

Oct. 2018

- Required to develop a system to simulate crossroad using pthreads. Given the list of cars and corresponding departure locations and destinations, the crossroad can work without a deadlock or car crashes.
- Use simple strategy of forcing the cars to acquire locks in the same order to prevent deadlock.

File System Based on Linux ext2 File System

Nov. 2018

- Designed a command line-based file system that can copy, add and delete files/directories, restore deleted files, and diagnose and repair virtual disks if it breaks down
- Required to build the whole program with only system header files. Created recursive functions for adding, deleting, and restoring files/directories to virtual disks. Designed shared helper functions like searching particular directories' location in disks, checking if a path is a valid path.(all directories are stored in the disk).

Virtual Memory System on Linux

Nov. 2018

- Required to code the page swapping algorithms and good understanding of structure of virtual memory.
- Designed a virtual memory system. Given the list of file names (about 10000) and limited memory, the system had a good lookup speed
- Support opt, fifo, clock, and lru page replacement algorithms
- Applied more complex data structures to implement paging algorithms so that look up, insert, delete can take less than $O(n)$ steps. (Use "linked list + hashtable" for handling the LRU algorithms)

Virtual Broad Game

Apr. 2018

- A broad game that supports at supports 1 or more players. Each player can quit and join at any time and only one player can make a move at each term.
- Use struct to handle properties of each player, each new player will be appended to the end of the user lists.
- Use socket to make connections between each user and carefully handle any possible failures like user quit after making a move or user quit after another user joins the game and so on.

Hijacking System Calls

Sep. 2018

- A script that can "hijack some system calls". By calling "add" with a certain "pid" and add process with "pid" into a monitored list. Any syscall from process in the monitored list can executed normally after the console logging some message about the syscall.
- Use spinlock to prevent race condition while updating the syscall table.
- Implement a new syscall that updates the monitored list and syscall table.
- "Hijack" the exit group function and replace it with a new one so that object in the monitored list corresponded to some process can be removed normally when a process exits.